The Role of Crossover in an Immunity Based Genetic Algorithm for Multimodal Function Optimization

Chien-Feng Huang
Modeling, Algorithms and Informatics Group (CCS-3)
Los Alamos National Laboratory, MS B256
Los Alamos, NM 87545
USA
cfhuang@lanl.gov

Abstract- When Genetic Algorithms are employed in multimodal function optimization, identifying multiple peaks and maintaining subpopulations of the search space are two central themes. In this paper, we use an immune system model to explore the role of crossover in GAs with respect to these two issues. The experimental results reported here will shed more light into how crossover affects the GA's search power in the context of multimodal function optimization. We will also show that an adaptive crossover strategy successfully achieves the two goals simultaneously. These results on the effects of crossover are a step toward a deeper understanding of how GAs work, and thus how to design more robust GAs for solving multimodal optimization problems.

1 Introduction

The process of information exchange among the population of individuals manipulated by Genetic Algorithms (GAs) [7] involves two key components: crossover and mate selection [9]. The investigation for the role of these two factors in advancing GA's search for a single, best-so-far solution,[1] has been conducted in [15] and [9], respectively. In this paper we aim at studying the role of crossover in GAs for multimodal function optimization.[2]

In the setting of multimodal function optimization, engineering and machine learning, there are two important issues when a GA is used: (1) how fast can a GA discover one or several peaks? And (2) can a GA maintain diverse subpopulations in different parts of the search space?[3]

Although considerable research has been conducted for the importance of crossover in terms of the GA's search for a single, better solution, the role of crossover is still not clear when GAs are used in the context of multimodal optimization.

In this paper we intend to employ an immune system model proposed by Smith, Forrest and Perelson [14], which was shown to be an effective algorithm for multimodal optimization, in order to advance our understanding for the role of crossover, and further design more robust GAs for practical tasks. Before delving fully into this paper, we briefly review Smith et al.'s immune system model and discuss how it facilitates formation of subpopulations over different areas of the search space. Section 3 illustrates empirical results for providing some answers to the two issues discussed above. Then we will extend the framework for studying the role of crossover to an adaptive crossover strategy, and show how this approach could accomplish the two goals simultaneously. Finally, this paper is concluded with the insights obtained and future research lines.

2 An Binary Immune System Model

Fitness sharing was an idea motivated by Holland's discussion [7] in which the number of individuals occupying a niche is limited to that niche's carrying capacity. Goldberg and Richardson [4] then introduced a fitness sharing mechanism that induces population diversity by penalizing individuals for the presence of similar individuals in the population. The technique they proposed was shown to be an effective method for maintaining subpopulations over several high-fitness regions of the search space. However, one of the serious limitations of this approach is that setting σ_s (a critical parameter in the fitness sharing scheme that represents a cutoff distance, beyond which no sharing will occur) requires knowledge about the number of peaks in the search space. This limitation arises from the fact that

[1] This metric is defined as the fitness of the best individual that has been seen thus far by generation n.

[2] See [10] for relevant research work conducted for the role of mate selection in GAs in the context of multimodal function optimization.

[3] The first issue was briefly discussed in [9]. For the second issue, there are some practical problems where maintaining subpopulations are critical. An example is the application of genetic approach to decentralized PI controller tuning for multivariable processes in [17].

fitness sharing is defined explicitly.

To avoid the difficulty of appropriately choosing σ_s Smith, Forrest and Perelson [14] proposed an algorithm that does not require explicit construction of the sharing function. Their approach can implicitly achieve fitness sharing that discovers for itself how many peaks are in the search space, and allocate trials appropriately. The idea is to use the metaphor of biological immune systems that can maintain the diversity needed for it to detect multiple antigens. Then the GA, along with the immune system algorithm, effectively distributes the population over several high-fitness areas of the search space.

The immune system model considered in this paper is based on a model introduced by Farmer et al. [1], where both antigens and antibodies are represented by binary strings. It is a simplification from the real biology in which genes are specified by a four-letter nucleic acid alphabet and recognition between antibodies and antigens is based on their three-dimensional shapes and physical properties. However, this abstract model of binary strings is rich enough for exploring how a relatively small number of recognizers (the antibodies) can evolve to recognize a much larger number of different patterns (the antigens).

In this binary immune system model, recognition is evaluated through a string matching procedure. The antigens are considered fixed, and a population of N antibodies is evolved to recognize the antigens using a GA. For any set of antigens, the goal is to obtain an antibody *cover*—a set of antibodies such that each antigen is recognized by at least one antibody in the population. Maintaining diverse antibodies is crucial for obtaining a cover [14].

An antibody is said to match an antigen if their bit strings are complementary (maximally different). Since each antibody may have to match against several different antigens simultaneously, we do not require perfect bit-wise matching. Many possible match rules are plausible physiologically (see [12] for examples). The degree of match is quantified by a class of match score functions $M : Antigen \times Antibody \rightarrow \Re$. For instance, M can simply count the number of complementary bits or M can identify contiguous regions of complementary bitwise matches within the string.

Smith et al. [14] adopted a model in which a fixed set of antigens is given, and the antibodies are initialized either to be completely random (to see if the GA can learn the correct antibodies) or initially given the answer by setting the population to include the correct antibodies (to test the stability of the answer). Their mechanism for fitness scoring is as follows:

1. A single antigen is randomly selected from the antigen population.
2. From the population of N antibodies a randomly selected sample of size σ is taken without replacement.
3. For each antibody in the sample, match it against the selected antigen, determine the number of bits that match, and assign it a match score.
4. The antibody in the sample population with the highest match score is determined. Ties are broken at random.
5. The match score of the winning antibody is added to its fitness. The fitness of all other antibodies remains unchanged.
6. This process is repeated for C cycles (typically one to three times the number of antibodies).

In this scoring scheme, since an antibody's fitness is increased only if it is the best matching antibody in the sample, the fitness values of antibodies are interdependent. In [14] Smith et al. showed analytically how this procedure implicitly embodies fitness sharing. Furthermore, Forrest et al. [2] reported that this scheme can maintain subpopulations of antibodies that cover a set of antigens.

Table 1: Illustration of the immune-based GAs.

1. Randomly generate an initial population of n antibodies.
2. Evaluate the fitnesses of antibodies by the six steps of Smith et al.'s algorithm.
3. Repeat until n offspring have been created.
   a. Select a pair of parents for mating.
   b. Apply crossover operator.
   c. Apply mutation operator.
4. Reset all the new individuals' fitnesses to zero and replace the current population with the new population.
5. Go to Step 2 until terminating condition.

The illustration of the GA system using Smith et al.'s algorithm is displayed in Table 1.[4]

[4]Since in Smith et al.'s algorithm the match scores of winning

Table 2: Building blocks of antigens

s_1 = 11111**********; c_1 =10
s_2 = *****11111**********; c_2 =10
s_3 = **********11111*****; c_3 =10
s_4 = **********11111; c_4 =10
s_5 = 00000**********; c_5 =10
s_6 = *****00000**********; c_6 =10
s_7 = **********00000*****; c_7 =10
s_8 = **********00000; c_8 =10

3 Experimental Results

To illustrate effects of crossover on Smith et al.'s immune algorithm (we call it the multimodal algorithm from here on), we use a simple example in which antigen populations cannot be matched by a single antibody type. Consider an antigen population that is composed of 50% $000\ldots0$ (all 0's) and 50% $111\ldots1$ (all 1's). In order for an antibody population to recognize these antigens, there would need to be some antibodies that are all 1's and others that are all 0's. Thus, a solution to this problem requires the GA to maintain two different solutions simultaneously. This is an example of a "multiple peaks" problem because there are two incompatible solutions that are maximally different. Typically, on multiple-peaks problems it is difficult for simple GAs to distribute the population over several peaks of a fitness landscape (two different subpopulations of antibodies that match two types of antigens, in this case). This is because the selection pressure in a simple standard GA usually entails strong convergence tendency to only one peak. Even without selection pressure, genetic drift due to sampling error can still lead the GA to converge on one of the peaks [5].

In light of pattern-recognition, Forrest et al. [2] pointed out that the immune system needs to recognize bacteria partially on the basis of the existence of certain unusual molecules that are inherently different from human cells, since many bacteria have cell walls made from polymers that do not occur in humans. With this as motivation, we study the GA's ability to detect common patterns (building blocks) in the antigen population and adopt the building-block idea in [7] to calculate fitnesses of antibodies.

Table 2 illustrates the building blocks of antigens $111\ldots1$ and $000\ldots0$ (string length is of 20 bits[5]). In this paper, an antibody is said to match an antigen if its bit string is complementary to the antigen at certain building blocks. Therefore, the match score function M_b is to identify the building blocks for which an antibody matches an antigen, and then assign corresponding scores to that antibody.

Specifically, the building-block-based function involves a set of schemata $S = \{s_1, \ldots, s_8\}$ and the fitness (match score) of a bit string x (antibody) is defined as

$$M_b(x) = \sum_{s_i \in S} c_i \delta_{s_i}(x),$$

where each c_i is a value assigned to the schema s_i as defined in the table; $\delta_{s_i}(x)$ is defined as 1 if x is an instance of the complement of s_i and 0 otherwise. For example, given an antigen $111\ldots1$, an antibody with the first five and the last five bits being all 0's will receive score $c_1 + c_4 = 20$, since these ten bits are complementary to those of the antigen.

Smith et al. [14] considered two cases for the score calculation of antibodies—perfect match and partial match. In case of perfect match, an antibody receives a non-zero score only if it perfectly matches the antigen. In case of partial match, an antibody receives a non-zero score if it partially matches the antigen. Therefore, the degree by which an antibody matches an antigen is indicated by the number of their complementary bits, and this determines the specificity of an antibody. Note that the consequence of a partial matching rule is that there is a trade-off between the number of antibodies used and their specificity—as the specificity of antibodies increases, so does the number of antibodies required to achieve a certain level of detection [6].

For the scoring rule discussed in this building-block-based recognition problem, we can also expand its definition by allowing partial match. That is, a prefect building-block match indicates that an antibody scores if all of its bits at a building block are complementary to those of an antigen. On the other hand, an example for partial match can allow an antibody to score with, say 80% bits (i.e., 4 bits in the case of the building blocks shown in Table 2) of a building block at which it matches an antigen. The result of this flexible scoring is a smaller population size required to achieve a certain level of recognition performance.[6] In this paper, we report the results obtained for the cases of the 80% building-block match.

antibodies are continuously accumulated, after each generation their fitness values can be large. Thus at step 4 of Table 1 we reset the fitnesses of the new population individuals to zero after each generation to prevent fitnesses from unlimited increase.

[5]The small string length here serves well for illustrating the effects of crossover. We currently have some results for larger string length that are consistent with the results obtained here.

[6]In the case of 100% building-block match, several experiments show similar qualitative results as the 80% building-block match case. But it requires much larger population size (i.e., higher computation cost) to achieve similar performance level.

Table 3: The mean function evaluations for discovering antibodies $111\ldots1$ and $000\ldots0$

Antibody	$111\ldots1$	$000\ldots0$
Crossover rate 0	4420 (777)	6940 (1299)
Crossover rate 0.3	2460 (286)	3420 (550)
Crossover rate 0.5	2500 (276)	2680 (330)
Crossover rate 0.7	2340 (369)	2300 (207)
Crossover rate 1	2180 (255)	1860 (212)

3.1 Effects on Discovery of Peaks

To address the question mentioned in the beginning of this paper we conduct a series of GA experiments using the multimodal algorithm. Our first objective is to investigate effects of one-point crossover on the peak-discovery capability of the immunity-based GA system, if any. Unless stated otherwise, these experiments use an antibody population size of 100, a binary tournament selection scheme [3],[7] one-point crossover with various rates, mutation rate of 0.005, and ran for 150 generations. The antigen population is 50% $000\ldots0$ and 50% $111\ldots1$, and both antigens and antibodies are binary strings of length 20. The number of samples, σ, is 10, which is 10% of the population size. We choose this value because Smith et al.'s analysis suggests that too small or too large a sample size cannot show fitness sharing's effect. In addition, as mentioned in [14], since the number of cycles (C) does not have a bearing on the antibodies' expected fitnesses, we use 100 cycles (i.e., population size) for each generation and this turns out to serve well for our investigation. Thus the number of the total function evaluations for each GA run are generations×cycles×sample size, which equals 150,000.

Table 3 displays the averaged mean function evaluations (over 50 runs) required for discovering $111\ldots1$ and $000\ldots0$ by the GA with crossover rate of 0, 0.3, 0.5, 0.7 and 1, respectively. (The numbers in parentheses are the corresponding standard errors.) These results show that as crossover rate increases, the GA tends to discover peaks with less time. This indicates that crossover can indeed enhance the GA's search power while locating multiple peaks, which is an extension of the study concerning the role of crossover in GA's search for a single, best-so-far solution.

[7]Tournament selection is employed here for low computational cost.

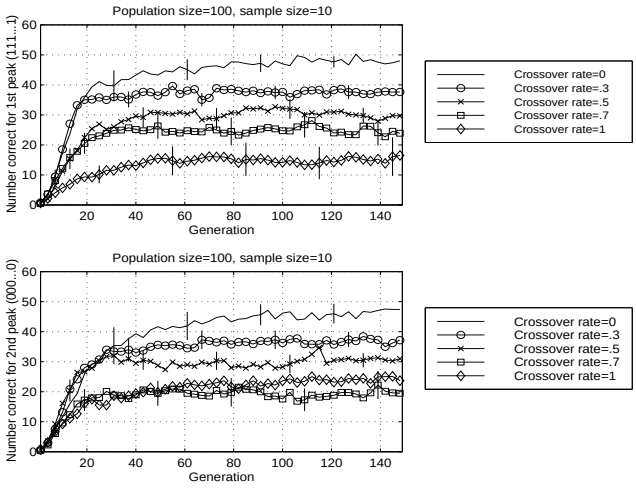

Figure 1: The number of antibodies that correctly recognize antigens

3.2 Effects on Maintaining Subpopulations

Once the desired antibodies are discovered, the next theme is to investigate the role of crossover in GAs with respect to maintaining these antibody subpopulations. Figure 1 displays the experimental results (averaged over 50 runs) for the number of antibodies that correctly recognize antigens.[8] One can see that the proportion of the correct antibodies maintained by the GAs using higher crossover rates tends to significantly drop. The reason is in the following: as crossover rate is increased, matings between the GA's population individuals would naturally increase the probability of producing useless hybrids (individuals that fall into the valley between the two peaks). E.g., given antibodies $111\ldots1$ and $000\ldots0$, higher crossover rate tends to increase the probability of the crossing-over between the two strings, which is then more likely to generate offspring that are away from the two peaks.

3.3 Adaptive Crossover Rates

The results illustrated thus far indicate that one would need to run the GAs with various crossover rates, depending on whether the goal is to identify multiple peaks, or to maintain multiple subpopulations. Then what if the two goals need to be fulfilled simultaneously?

To answer this question, we propose an adaptive crossover strategy—at the first generation, crossover rate is set at 1, and gradually declined by a certain

[8]The vertical bars overlapping the curves for the number of the desired antibodies are 95% confidence intervals calculated from Student's t-statistic [11].

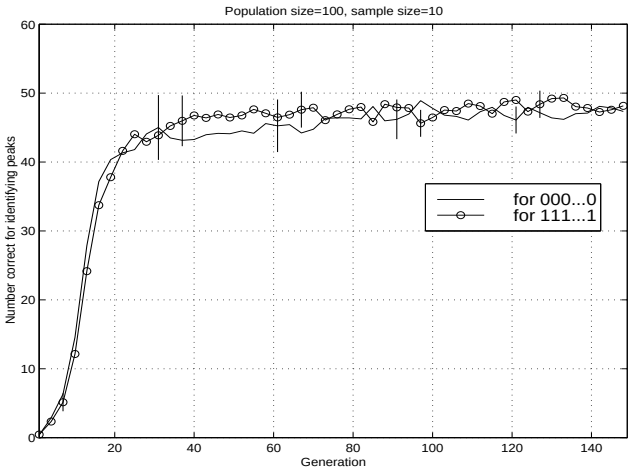

Figure 2: The number of antibodies that correctly recognize antigens based on the adaptive crossover strategy

amount at each consecutive generation until it reaches 0. Then the crossover rate is kept at 0 for all the remaining generations. In this paper we allow crossover rate to decline by 0.1 at every generation until it reaches 0. Thus the crossover rate is 1 at the first generation, and is 0.9 at generation 2, and so on. Then the GA will use crossover rate of 0 for generations after 10.

The empirical results for the GAs using the adaptive crossover scheme show that the averaged function evaluations (over 50 runs) for discovering $111\ldots1$ and $000\ldots0$ is 3000 and 3080, respectively (the standard error is 317 and 848, respectively). In comparison with the results obtained for crossover rate 0 in Table 3, one can see that adapting crossover rates can reduce the time required for discovering multiple peaks. On the other hand, compared to the results obtained for crossover rate 1 in Figure 1, Figure 2 shows that the number of desired antibodies has significantly increased, as well. Although this framework of the adaptive crossover strategy is rather simple, it indeed sheds some light into how the two goals can be achieved simultaneously.

3.4 Effects of Population Size

In Table 3, one may notice that the GA using crossover rate of 1 can discover the peaks in only about two generations; and in case of crossover rate of 0, the GA requires only about four to seven generations to do so. This is because the multimodal testbed considered here is rather simple, and a population of 100 antibodies seems adequate for the discover of the peaks. In practical situations where the size of the GA population may not be sufficient for searching a given problem space, one may wonder how crossover would affect the GA's power in discovering numerous peaks and maintaining subpopulations.

Table 4: The number of runs (out of 50) in which antibodies $111\ldots1$ and $000\ldots0$ are discovered

Antibody	$111\ldots1$	$000\ldots0$
Crossover rate 0	48	50
Crossover rate 0.3	49	49
Crossover rate 0.5	48	46
Crossover rate 0.7	47	49
Crossover rate 1	49	50

In this subsection, we present empirical results for studying the effects of population size in the immune GA model. Table 4 illustrates the results for the number of runs (out of 50) in which antibodies $111\ldots1$ and $000\ldots0$ are discovered, respectively, based on population size 50 and sample size 5 (other parameter values remain unchanged). One can see that in almost each case there is one or few runs where the antibodies are not discovered. A closer inspection indicates that the founder effect has seriously constrained the GA's search for these peaks. In GA research, the founder effect has been identified as an important factor that hampers the GA's search process [8]—in presence of incompatible schemata, the first discovered of the incompatible schemata comes to occupy a large portion of the population, and constrains future evolutionary progress. Consequently, the founder schema effectively precludes the testing of the other incompatible schema. Further improvements stem from the founder, making it progressively less likely that the other schema will influence the search process.

We can observe the founder effect directly by plotting the densities (percentage of the population that are instances) of the relevant schemata over time for the GAs. Figure 3 displays the schema density dynamics for a typical run where antibody $111\ldots1$ is not discovered over the whole search course. Although each of s_1 and s_2 has instances at the first few generations, the instances of s_2 quickly found their dynasties and take over the whole population. As a consequence, future improvements are constrained by s_2 such that s_1 is prohibited from being further tested. (Due to sampling errors, either of each pair of the incompatible schemata is likely to dominate the population. Therefore, for other runs different schemata may take over the whole population.)

Table 5 displays the mean function evaluations for

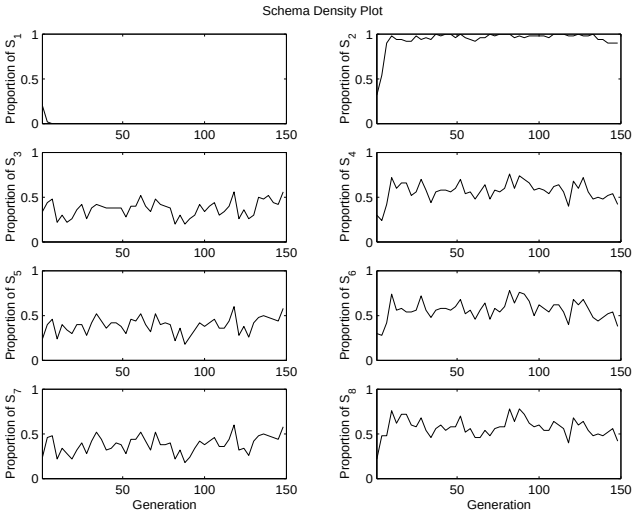

Figure 3: Schema dynamics for observing the founder effect

Table 5: The mean function evaluations for discovering antibodies $111\ldots1$ and $000\ldots0$

Antibody	$111\ldots1$	$000\ldots0$
Crossover rate 0	5208 (1002)	5770 (1053)
Crossover rate 0.3	1316 (244)	2071 (368)
Crossover rate 0.5	1089 (183)	1239 (181)
Crossover rate 0.7	814 (94)	1107 (172)
Crossover rate 1	960 (114)	948 (106)

discovering $111\ldots1$ and $000\ldots0$, averaged over the number of the runs in which they are located. Compared to the results shown in Table 3, one can see that as population size decreases, higher crossover rates tend to provide more advantage in discovering multiple peaks.

Figure 4 illustrates the corresponding results for the number of antibodies that correctly recognize antigens. In comparison with the results shown in Figure 1, one can see that the performance discrepancy between various crossover rates in maintaining subpopulations has become more indistinguishable.

Figure 5 reveals a typical run for the number of desired antibodies conducted on crossover rate of 1. This figure shows that $111\ldots1$ is drown out by $000\ldots0$ before generation 20, although both of them do show up in earlier generations. Further inspection shows that there are 24 (out of 50) runs in which most of the individuals converge to all 1's, and in 14 (out of 50) runs most of the individuals converge to all 0's, and there are 12 runs in which the two peaks are lost. Due to the insufficient population size, this kind of strong convergence occurs to all the cases tested here, which in turn generates the larger error bars displayed in Figure 4. Furthermore, as the population converges on one peak, the crossing-over between them will not generate many distinct hybrids that fall into the valley between the two peaks. One can thus see that higher crossover rates would not reduce the proportion of the correct antibodies maintained by the GAs. In other words, when the population size is not large enough, lower crossover rates will not facilitate maintaining subpopulations.

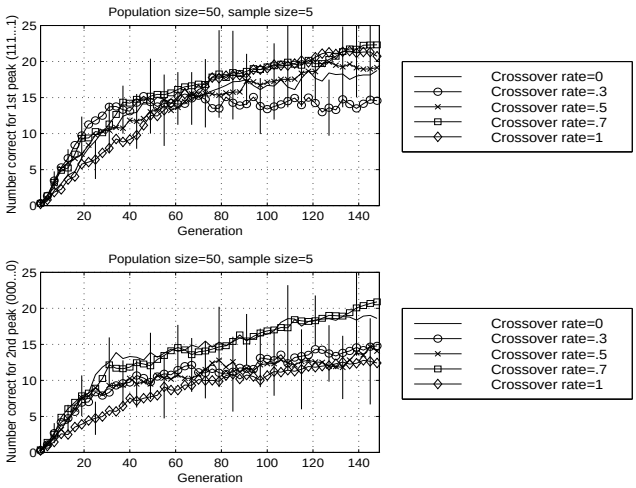

Figure 4: The number of antibodies that correctly recognize antigens

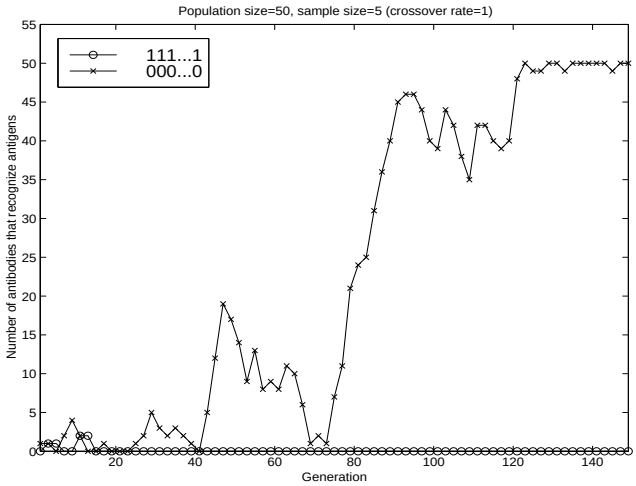

Figure 5: The number of antibodies that correctly recognize antigens (based on crossover rate 1)

4 Conclusion and Future Work

In this paper, we have described Smith et al.'s immune system model in which subpopulations can be maintained through specific interactions among antibodies and antigens. We have investigated the role of crossover in the immunity-based GA systems with respect to discovering multiple peaks and maintaining subpopulations. Both of these issues are important in the setting of multimodal function optimization.

In studying the peak-identifying problem, we showed that higher crossover rate facilitates discovering multiple peaks of the fitness landscape. This is a crucial extension for the research work that indicates crossover can advance the GA's search for a single, best-so-far solution. However, in studying the subpopulation-maintaining problem, the results show that higher crossover rate is harmful because the proportion of the antibodies that correctly recognize antigens is decreased. To resolve the conflict, we then propose an adaptive crossover strategy and show that it could fulfill the two goals simultaneously.

We have also studied the effects of population size on the immune GA system. As population size is not adequate for searching a problem space, the immune system model's performance in discovering peaks and maintaining subpopulations is degraded. The results reveal that, due to the founder effect, the GAs cannot always locate the peaks of the fitness landscape, even with high crossover rates. However, in the cases where peaks can be identified, higher crossover rates can still improve the GA's search speed for the peaks.

Since the pattern-recognition strategy in our approach was based on schema detection, it is worth further exploration because in real problems when there are many more antigens than antibodies, antibodies need to detect common regions. Our future work will extend the results for schema detection, identification of multiple peaks, and maintenance of subpopulations to more realistic scale of antigens and antibodies. On the other hand, since the results obtained for adaptive crossover strategies are encouraging, another research line is to investigate other plausible adaptive crossover schemes, such as the adaptive crossover distribution mechanism in [13], in order to develop more robust GAs for multimodal optimization tasks.

We have been concentrating on the study of the role of one-point crossover, our future work will extend to the study for two-point crossover and uniform crossover [16]. Finally, we would like to develop an analytical tool to enhance the understanding for the role of crossover in GAs when used in multimodal function optimization.

Bibliography

[1] Farmer, J. D., Packard, N. H., and Perelson, A. S.: The Immune System, Adaptation, and Machine Learning. In D. Farmer, A. Lapedes, N. Packard, and B. Wendroff (Eds.): Evolution, Games and Learning. NorthHolland (1986). (Reprinted from Physica, 22D, 187-204)

[2] Forrest, S., Javornik, B., Smith, R. E., and Perelson, A. S.: Using Genetic Algorithms to Explore Pattern Recognition in the Immune System. Evolutionary Computation, 1(3) (1993) 191-211.

[3] Goldberg, D. E. and Deb, K.: A Comparative Analysis of Selection Schemes used in Genetic Algorithms. Foundation of Genetic Algorithms (1991) 69-93.

[4] Goldberg, D. E. and Richardson, J.: Genetic Algorithms with Sharing for Multimodal Function Optimization. Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms (1987) 41-49.

[5] Goldberg, D. E. and Segrest, D.: Finite Markov Chain Analysis of Genetic Algorithms. International Conference on Genetic Algorithms, 2 (1987) 1-8.

[6] Hofmeyr, S. A., and Forrest, S.: Architecture for an Artificial Immune System. Evolutionary Computation, 8(4) (2000) 443-473.

[7] Holland, J. H.: Adaptation in Natural and Artificial Systems. Ann Arbor, MI: University of Michigan Press (1975).

[8] Holland, J. H.: Building Blocks, Cohort Genetic Algorithms, and Hyperplane-Defined Functions. Evolutionary Computation, 8(4) (2000) 373-391.

[9] Huang, C.-F.: A Study of Mate Selection in Genetic Algorithms. Doctoral dissertation. Ann Arbor, MI: University of Michigan, Electrical Engineering and Computer Science (2002).

[10] Huang, C.-F.: Using an Immune System Model to Explore Mate Selection in Genetic Algorithms. Proceedings of 2003 Genetic and Evolutionary Computation Conference (2003), in press.

[11] Miller, R. G.: Beyond ANOVA, Basics of Applied Statistics. John Wiley and Sons (1986).

[12] Perelson, A. S.: Immune Network Theory. Immunol. Rev., 110 (1989) 5-36.

[13] Schaffer, J. D. and Morishima, A.: An adaptive crossover Distribution Mechanism for Genetic Algorithms. Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms (1987).

[14] Smith, R., Forrest, S., and Perelson, A. S.: Searching for Diverse, Cooperative Populations with Genetic Algorithms. Evolutionary Computation, 1(2) (1993) 127-149.

[15] Spears, W.: Evolutionary Algorithms: The Role of Mutation and Recombination. Springer-Verlag (2000).

[16] Syswerda, G.: Uniform Crossover in Genetic Algorithms. Proceedings of the Third International Conference on Genetic Algorithms (1989).

[17] Vlachos, C., Williams, D., and Gomm, J. B.: Genetic Approach to Decentralized PI Controller Tuning for Multivariable Processes. IEE Proc. Control Theory and Applications, 146 (1999), 58-64.